UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/679,076	10/04/2003	Paul E. McKenney	BEA920030004US1	3528

49474 7590 03/25/2008
LAW OFFICES OF MICHAEL DRYJA
1474 N COOPER RD #105-248
GILBERT, AZ 85233

EXAMINER	
BERTRAM, RYAN	

ART UNIT	PAPER NUMBER
2187	

MAIL DATE	DELIVERY MODE
03/25/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/679,076	MCKENNEY ET AL.	
	Examiner	Art Unit	
	RYAN BERTRAM	2187	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 January 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15 and 17-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15 and 17-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *04 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Examiner acknowledges receipt of the Request for Continued Examination dated 1/21/2008 in response to the Advisory Action dated 12/10/2007. At this point, claims 1, 3-9, 12, and 18 have been amended, and claim 16 has been cancelled. Thus, claims 1-15 and 17-20 are now pending in the application.

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 1/21/2008 has been entered.

I. REJECTIONS BASED ON PRIOR ART

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 1-6, 8-10, and 18-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rajwar et al. (US 7,120,762) hereinafter Rajwar, in view of Herlihy et al. (Transactional Memory: Architectual Support for Lock-Free Data Structures) hereinafter Herlihy.

1. Regarding claim 1, Rajwar teaches a method comprising:

executing a code section relating to memory using a hardware transactional approach to executing code sections **[see Col. 2, lines 30-36; critical section is speculatively executed without the use of software locks]**, and

where the hardware transactional approach to executing code sections fails a threshold in executing the code section, executing the code section relating to the memory using a software approach to locking memory **[see Col. 3, lines 60-64; critical section re-executed a predetermined number of times. If a conflict still exists, a lock variable is acquired and used]**

Rajwar does not expressly disclose that the hardware transactional approach is implemented in hardware other than a processor.

Herlihy discloses a system that executes code relating to transactional memory in hardware other than a processor, specifically a transactional memory cache **[see Page 3, section 3.1.1]**.

Rajwar and Herlihy are analogous art because they are from the same field of endeavor, namely executing code relating to transactional memory.

At the time of the invention it would have been obvious to a person of ordinary skill in the art modify the system of Rajwar with the transactional cache of Herlihy.

The motivation for doing so would have been to make lock-free synchronization as efficient as conventional techniques **[see abstract]**.

Therefore, it would have been obvious to combine Rajwar with Herlihy for the benefit of efficient lock-free synchronization, to obtain the invention as specified in claim 1.

2. Regarding claim 2, Rajwar and Herlihy teach the method of claim 1, further comprising executing other code sections relating to the memory using the software

approach to locking memory **[see Col. 3, lines 60-64; the critical section of any code is executed using the software approach if the hardware approach fails a predetermined number of times]**.

3. Regarding claim 3, Rajwar and Herlihy teach the method of claim 1, wherein executing the code section relating to the memory using the hardware transactional approach to executing code sections comprises:

starting a transaction inclusive of the code section **[see Col. 2, lines 30-36; critical section begins execution]**; and,

conditionally executing the transaction **[see Col. 7, lines 44-50; critical section conditionally executed]**.

4. Regarding claim 4, Rajwar and Herlihy teach the method of claim 3, wherein executing the code section relating to the memory using the hardware transactional approach to executing code sections section further comprises, upon successfully completing the transaction, committing execution of the transaction to the memory to which the code section relates **[see Col. 2, lines 30-36; execution committed if no errors occur]**.

5. Regarding claims 5 and 6, Rajwar and Herlihy teach the method of claim 1, wherein the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises the hardware transactional approach

to executing code sections aborting execution of the code section a single time or a predetermined number of times **[see Col. 3, lines 60-64, the retry limit can be set at any number]**.

6. Regarding claim 8, Rajwar and Herlihy teach the method of claim 1, wherein the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises utilizing information passed from a compiler to determine whether to utilize the software approach to locking memory to execute the code section **[see Col. 3, lines 1-5; existing compiler detects critical section]**.

7. Regarding claim 9, Rajwar and Herlihy teach the method of claim 1, wherein the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises tracking a success rate of the hardware transactional approach to executing code sections in executing the code section to determine whether to utilize the software approach to locking memory to execute the code section **[see Col. 8, lines 12-15; prediction table with success rate used to determine if software approach should be used]**.

8. Regarding claim 10, Rajwar and Herlihy teach the method of claim 1, wherein executing the code section relating to the memory using the software approach to locking memory comprises:

placing a lock on the memory to which the code section relates **[see Fig 2, 30, Col. 5, lines 17-19; lock acquired]**;

executing the code section **[see Fig. 2, 44, Col. 5, lines 23-25; critical section executed]**;

committing execution of the code section to the memory as the code section is executed; and **[see Fig. 2, 44, Col. 5, lines 23-25; critical section executed, changes are committed]**,

removing the lock on the memory to which the code section relates **[see Fig. 2, 32, Col. 5, lines 24-26; lock released]**.

9. Regarding claim 18, Rajwar and Herlihy teach an article of manufacture comprising:

a tangible computer-readable medium **[see Fig. 1, 17; shared memory]**; and, means in the medium for executing a code section using a software approach to locking memory where a hardware transactional approach to executing code sections has failed a threshold in executing the code section **[see Col. 2, lines 30-36; Col. 3, lines 60-64; critical section executed using software lock if speculative hardware approach fails]**, the hardware approach implemented in hardware other than a processor **[see Herlihy page 3, section 3.1.1; transactional memory cache]**.

10. Regarding claim 19, Rajwar and Herlihy teach the article of claim 18, wherein the means executes the code section using the software approach to locking memory

where the hardware transactional approach to executing code sections has failed to execute the code section a predetermined one or more times **[see Col. 3, lines 60-64; critical section re-executed a predetermined number of times. If a conflict still exists, a lock variable is acquired and used]**.

11. Regarding claim 20, Rajwar and Herlihy teach the article of claim 18, wherein the tangible computer-readable medium is a recordable data storage medium **[see Fig. 1, 17; shared memory device]**.

Claims 12-15 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rajwar in view of Herlihy as applied to claims 1-6, 8-10, and 18-20 above, and in further view of Radovic et al (US 2004/0098723) hereinafter Radovic.

12. Regarding claim 12, Rajwar and Herlihy teach a system comprising:

a processor **[see Fig. 1, 16; processor]**; and,

a memory storing a lock function to execute a code section via a software approach to locking memory where the hardware transactional approach to executing code sections has initially failed a threshold in executing the code section **[see Fig. 1, 17; Col. 2, lines 30-36; Col. 3, lines 60-64; critical section executed using software lock if speculative hardware approach fails]**,

wherein the hardware transactional approach to executing code sections is implemented in hardware other than the processor **[see Herlihy, Page 3, section 3.1.1; transactional memory cache]**.

Rajwar and Herlihy do not disclose expressly that the locking function is a spin lock function.

Radovic discloses a multiprocessing system that employs a variety of software locks, including one of a spin lock **[see paragraph 0008]**.

Rajwar, Herlihy, and Radovic are analogous art because they are from the same field of endeavor, namely executing code relating to transactional memory.

At the time of invention it would have been obvious to one having ordinary skill in the art to modify the system of Rajwar and Herlihy with the spin lock taught by Radovic.

The motivation for doing so would have been to ensure that only one process at a time could access a critical section of memory **[see Radovic, paragraph 0008]**.

Therefore it would have been obvious to combine Radovic and Herlihy with Rajwar for the benefit of preventing multiple process accesses to a critical section of memory to obtain the invention as specified in claim 12.

13. Regarding claim 13, Rajwar, Herlihy, and Radovic teach the system of claim 12, further comprising a plurality of nodes interconnected to one another, one of the plurality of nodes inclusive of the processor and the memory **[see Rajwar, Fig. 1; processor unit 12 includes processor 16 and memory]**.

14. Regarding claim 14, Rajwar, Herlihy, and Radovic teach the system of claim 12, wherein the memory further stores the code section, the code section programmed to call the spin lock function to execute, the spin lock function locking a portion of the memory to which the code section relates **[see Radovic, paragraphs 0008 & 0026; memory stores instruction code, including the spin lock function]**.

15. Regarding claim 15, Rajwar, Herlihy, and Radovic teach the system of claim 14, wherein the memory further stores a spin unlock function that the code section calls to unlock the portion of the memory to which the code section relates **[see Radovic, paragraphs 0008 & 0026; memory stores instruction code, including the spin unlock function]**.

16. Regarding claim 17, Rajwar, Herlihy, and Radovic teach the system of claim 12, wherein the software approach is used to execute the code section where the hardware transactional approach to executing code sections has failed to execute the code section a predetermined one or more times **[see Rajwar, Col. 3, lines 60-64; critical**

section re-executed a predetermined number of times. If a conflict still exists, a lock variable is acquired and used].

Claims 7 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rajwar in view of Herlihy, as applied to claims 1-6, 8-10, and 18-20 above, and in further view of McKenney (US 2002/0078284).

17. Regarding claim 7, Rajwar and Herlihy teach the method of claim 1, wherein utilization of the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises using a success rate in determining whether to utilize the software approach to locking memory to execute the code section **[see above]**.

Rajwar and Herlihy do not disclose expressly that a digital filter is used as the threshold for determining whether to use the software approach.

McKenney discloses a method for dynamically selecting an optimal lock mode in a multiprocessor environment. He also discloses using a digital filter for taking measurements to determine an optimal opportunity for switching to an alternative locking mode **[see paragraphs 9 & 29]**.

Rajwar, Herlihy, and McKenney are analogous art because they are from the same field of endeavor, namely executing code relating to transactional memory.

At the time of invention it would have been obvious to a person of ordinary skill in the art to modify the system of Rajwar and Herlihy with the digital filter taught by McKenney.

The motivation for doing so would have been to determine the optimal opportunity for switching to an alternative locking mode **[see McKenney, paragraph 29]**.

Therefore, it would have been obvious to combine Rajwar and Herlihy with Radovic for the benefit of efficiently determining when to switch to a different locking mode to obtain the invention as specified in claim 7.

18. Regarding claim 11, Rajwar and Herlihy teach the method of claim 1 as described above.

Rajwar and Herlihy do not disclose that executing the code section relating to the memory using the software approach to locking memory comprises utilizing one of a queued lock, a non-uniform memory architecture (NUMA) lock, and a reader-writer lock.

McKenney discloses the lock mode in NUMA machines consisting of a reader-writer lock mode **[see paragraph 9; reader-writer lock]**.

At the time of invention it would have been obvious to a person of ordinary skill in the art to modify the system of Rajwar and Herlihy with software locking options taught by McKenney.

The motivation for doing so would have been to synchronize processes **[see paragraph 9]**.

Therefore, it would have been obvious to combine Rajwar and Herlihy with Radovic for the benefit synchronized processes, to obtain the invention as specified in claim 11.

Claims 1-6, 8, and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tremblay et al. (US 2004/0162967) hereinafter Tremblay in view of Herlihy.

19. Regarding claim 1, Tremblay discloses a method comprising: utilizing a hardware transactional approach to executing code sections to execute a code section relating to memory **[see paragraph 67-68; code in critical section transactionally executed]**;

and, where utilizing the hardware transactional approach to executing code sections fails a threshold in executing the code section, utilizing a software approach to locking memory to execute the code section relating to the memory **[see paragraph 70; if re-attempts fail, critical section executed using software locks]**.

Tremblay does not expressly disclose that the hardware approach is implemented in hardware other than a processor.

Herlihy discloses a system that that executes code relating to transactional memory in hardware other than a processor, specifically a transactional memory cache **[see Page 3, section 3.1.1]**.

Tremblay and Herlihy are analogous art because they are from the same field of endeavor, namely executing code relating to transactional memory.

At the time of the invention it would have been obvious to a person of ordinary skill in the art modify the system of Trembay with the transactional cache of Herlihy.

The motivation for doing so would have been to make lock-free synchronization as efficient as conventional techniques **[see abstract]**.

Therefore, it would have been obvious to combine Tremblay with Herlihy for the benefit of efficient lock-free synchronization, to obtain the invention as specified in claim 1.

20. Regarding claim 2, Tremblay and Herlihy disclose the method of claim 1, further comprising utilizing the software approach to locking memory to execute other code sections relating to the memory **[see paragraph 70; the critical section of any code is executed using the software approach if the hardware approach fails a predetermined number of times]**.

21. Regarding claims 3 and 4, Tremblay and Herlihy disclose the method of claim 1, wherein utilizing the hardware transactional approach to executing code sections to execute the code section comprises: starting a transaction inclusive of the code section **[see paragraph 64; start transactional execution (STE)]**; and, conditionally executing the transaction **[see paragraph 64; changes are committed if no errors occur]**; and the method of claim 3, wherein utilizing the hardware transactional approach to executing code sections to execute the code section further comprises, upon successfully completing the transaction, committing execution of the transaction to the memory to which the code section relates **[see paragraphs 64 and 67; changes are committed to memory if no errors occur]**.

22. Regarding claims 5 and 6, Tremblay and Herlihy disclose the method of claim 1, wherein utilization of the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises the hardware transactional approach to executing code sections aborting execution of the code section a single time **[see paragraph 70; fail threshold retries can be zero, one, or two or more times]**; and the method of claim 1, wherein utilization of the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises the hardware transactional approach to executing code sections aborting execution of the code section a predetermined plurality of times **[see paragraphs 70; fail threshold retries can be zero, one, or two or more times]**.

23. Regarding claim 8, Tremblay and Herlihy disclose the method of claim 1, wherein utilization of the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises utilizing information passed from a compiler to determine whether to utilize the software approach to locking memory to execute the code section **[see paragraph 65; compiler determines if lock is necessary]**.

24. Regarding claim 10, Tremblay and Herlihy disclose the method of claim 1, wherein utilizing the software approach to locking memory to execute the code section comprises: placing a lock on the memory to which the code section relates; executing the code section; committing execution of the code section to the memory as the code

section is executed; and, removing the lock on the memory to which the code section relates **[see paragraph 70;]**.

Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Tremblay in view of Herlihy as applied to claim 1 above, and further in view of Rajwar.

25. Regarding claim 9, Tremblay and Herlihy disclose all of the limitations of claim 1, as discussed above. However, he does not disclose using a success rate to determine whether the software approach to locking memory should be used.

Rajwar teaches a method for speculatively executing critical sections of multi-threaded programs. He discloses using a table with values of historical success of previous execution of certain critical sections **[see paragraph 55]** in order to prevent using a hardware approach to execution if the critical sections will be more likely to be contested with **[see paragraph 21]**.

Tremblay, Herlihy, and Rajwar are analogous art because they are from the same field of endeavor of using software locks to execute a critical section of code.

At the time of the invention it would have been obvious to a person of ordinary skill in the art modify the system of Trembay and Herlihy with success rate tracking of Rajwar.

The motivation for doing so would have been to be able to determine whether the hardware or software approach should be used **[see Col. 8, lines 12-20]**.

Therefore, it would have been obvious to combine Tremblay and Herlihy with Rajwar for the benefit of being able to determine whether the hardware or software approach should be used, to obtain the invention as specified in claim 9.

Claims 7 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tremblay in view of Herlihy as applied to claim 1 above, and further in view of McKenney (US 2002/0078284).

26. Regarding claim 7, Tremblay and Herlihy teach the method of claim 1, wherein utilization of the hardware transactional approach to executing code sections failing the threshold in executing the code section comprises using a success rate in determining whether to utilize the software approach to locking memory to execute the code section **[see above]**.

Tremblay and Herlihy do disclose expressly that a digital filter is used as the threshold for determining whether to use the software approach.

McKenney discloses a method for dynamically selecting an optimal lock mode in a multiprocessor environment. He also discloses using a digital filter for taking measurements to determine an optimal opportunity for switching to an alternative locking mode **[see paragraphs 9 & 29]**.

Tremblay, Herlihy, and McKenney are analogous art because they are from the same field of endeavor, namely executing code relating to transactional memory.

At the time of invention it would have been obvious to a person of ordinary skill in the art to modify the system of Tremblay and Herlihy with the digital filter taught by McKenney.

The motivation for doing so would have been to determine the optimal opportunity for switching to an alternative locking mode **[see McKenney, paragraph 29]**.

Therefore, it would have been obvious to combine Tremblay and Herlihy with Radovic for the benefit of efficiently determining when to switch to a different locking mode to obtain the invention as specified in claim 7.

27. Regarding claim 11, Tremblay and Herlihy teach the method of claim 1 as described above.

Tremblay and Herlihy do not disclose that executing the code section relating to the memory using the software approach to locking memory comprises utilizing one of a queued lock, a non-uniform memory architecture (NUMA) lock, and a reader-writer lock.

McKenney discloses the lock mode in NUMA machines consisting of a reader-writer lock mode **[see paragraph 9; reader-writer lock]**.

At the time of invention it would have been obvious to a person of ordinary skill in the art to modify the system of Tremblay and Herlihy with software locking options taught by McKenney.

The motivation for doing so would have been to synchronize processes **[see paragraph 9]**.

Therefore, it would have been obvious to combine Tremblay and Herlihy with Radovic for the benefit synchronized processes, to obtain the invention as specified in claim 11.

Claim 12-15 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tremblay in view of Herlihy, and further in view of Radovic et al. (US 2004/0098723) hereafter Radovic.

28. Regarding claim 12, Tremblay and Herlihy teach a system comprising:

a processor **[see paragraph 53; processor]**; and,

a memory **[paragraph 53; main memory]** storing a lock function to execute a code section via a software approach to locking memory where the hardware transactional approach to executing code sections has initially failed a threshold in executing the code section **[see paragraphs 68-70; critical section executed using software lock if speculative hardware approach fails]**,

wherein the hardware transactional approach to executing code sections is implemented in hardware other than the processor **[see Herlihy, Page 3, section 3.1.1; transactional memory cache]**.

Tremblay and Herlihy do not disclose expressly that the locking function is a spin lock function.

Radovic discloses a multiprocessing system that employs a variety of software locks, including one of a spin lock/unlock **[see paragraph 0008]**.

Tremblay, Herlihy, and Radovic are analogous art because they are from the same field of endeavor, namely executing code relating to transactional memory.

At the time of invention it would have been obvious to one having ordinary skill in the art to modify the system of Tremblay and Herlihy with the spin lock taught by Radovic.

The motivation for doing so would have been to ensure that only one process at a time could access a critical section of memory **[see Radovic, paragraph 0008]**.

Therefore it would have been obvious to combine Radovic and Herlihy with Tremblay for the benefit of preventing multiple process accesses to a critical section of memory to obtain the invention as specified in claim 12.

29. Regarding claim 13, Tremblay, Herlihy, and Radovic teach the system of claim 12, further comprising a plurality of nodes interconnected to one another, one of the plurality of nodes inclusive of the processor and the memory **[see Tremblay, Fig. 1; processor unit 101 includes processor and memory]**.

30. Regarding claim 14, Tremblay, Herlihy, and Radovic teach the system of claim 12, wherein the memory further stores the code section, the code section programmed to call the spin lock function to execute, the spin lock function locking a portion of the

memory to which the code section relates **[see Radovic, paragraphs 0008 & 0026; memory stores instruction code, including the spin lock function]**.

31. Regarding claim 15, Tremblay, Herlihy, and Radovic teach the system of claim 14, wherein the memory further stores a spin unlock function that the code section calls to unlock the portion of the memory to which the code section relates **[see Radovic, paragraphs 0008 & 0026; memory stores instruction code, including the spin unlock function]**.

32. Regarding claim 17, Tremblay, Herlihy, and Radovic teach the system of claim 12, wherein the software approach is used to execute the code section where the hardware transactional approach to executing code sections has failed to execute the code section a predetermined one or more times **[see Tremblay, paragraph 70; critical section re-executed a predetermined number of times. If a conflict still exists, a lock variable is acquired and used]**.

Claim 18-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tremblay in view of Herlihy as applied to claim 1 above, and further in view of Rubinstein (US 5,913,215).

33. Claims 18-20 are method versions of claims 1 and 6, enabled by instructions rather than hardware. Tremblay discloses an apparatus for performing the method of claims 18-20, as discussed above in the rejection of claims 1 and 6.

Tremblay and Herlihy, however, do not expressly disclose that the method is performed by a software series of instructions, instead disclosing a set of hardware components.

Rubinstein discloses, on Col. 10, lines 3-15, that computer methods may be performed either by a series of instructions, or by specific hardware components that contain hard-wired logic for performing the method, or by any combination of the two.

Tremblay, Herlihy, and Rubinstein are analogous art because they are from the same general field of endeavor, namely computer-controlled methods.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to modify the apparatus of Tremblay and Herlihy by embodying it in executable instructions.

The motivation for doing so is portability and ease of installation. For example, it is well known that a method encoded in a program may be installed onto different systems much more quickly and easily than can hardware components designed to perform the same method.

Therefore, it would have been obvious to combine Tremblay and Herlihy with Rubinstein for the benefits shown above, to obtain the invention as specified in claims 18-20.

II. RESPONSE TO APPLICANT'S ARGUMENTS

1st Point of Argument

In response to applicant's argument that the combined prior art does not teach executing a code section using a hardware approach, the examiner directs applicant's attention to Col. 2, lines 30-36 of Rajwar and paragraph 68 of Tremblay. In both instances, it is taught that the code section is specutively executed, and all changes are committed only if no errors have occurred. This is the hardware transactional approach to executing a code section. This approach is analogous to the hardware approach taught in applicant's disclosure.

2nd Point of Argument

In response to applicant's arguments that the Herlihy reference is not properly considered prior art in the present application, applicant is invited to evaluate the dates of the present application and Herlihy's once again. Applicant has shown the effective date of Herlihy to be May of 1993. Even if the Rule 131 declaration was proper, giving the present application an effective date of January 30, 2003, *the Herlihy reference was*

published more than 9 years ago. Thus, the Herlihy reference is properly considered prior art.

3rd Point of Argument

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See In re Fine, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and In re Jones, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, Herlihy discloses an alternative way of handling transactional memory using a transactional memory cache. Using this technique would make lock-free synchronization as efficient as conventional techniques. This benefit serves as proper motivation for combining the references.

III. CLOSING COMMENTS

Conclusion

(a) **Status of Claims In the Application**

(i) Claims Rejected In the Application

Per the instant office action, claims 1-15 and 17-20 have received a fourth action on the merits and are subject of a non-final rejection.

(b) **Directions of Future Correspondences**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ryan Bertram whose telephone number is 571-270-1377. The examiner can normally be reached on Mon-Fri 9am-5:30pm ET.

Important Note

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Donald Sparks can be reached on 571-272-4201. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Donald Sparks/

Supervisory Patent Examiner, Art

Unit 2187

/Ryan Bertram/

Examiner, Art Unit 2187